

```

FFFFFFFFFFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFFFFFFFFFFF
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA

```

[illegible]

| | | | | | | | | | | | |
|----------|----|------------|----|---------|----|----------|----|----------|--|----------|--|
| DDDDDDDD | | AAAAAA | | PPPPPPP | | DDDDDDDD | | EEEEEEEE | | FFFFFFFF | |
| DDDDDDDD | | AAAAAA | | PPPPPPP | | DDDDDDDD | | EEEEEEEE | | FFFFFFFF | |
| DD | DD | AA | AA | PP | PP | DD | DD | EE | | FF | |
| DD | DD | AA | AA | PP | PP | DD | DD | EE | | FF | |
| DD | DD | AA | AA | PP | PP | DD | DD | EE | | FF | |
| DD | DD | AA | AA | PP | PP | DD | DD | EE | | FF | |
| DD | DD | AA | AA | PPPPPPP | | DD | DD | EEEEEEEE | | FFFFFFFF | |
| DD | DD | AA | AA | PPPPPPP | | DD | DD | EEEEEEEE | | FFFFFFFF | |
| DD | DD | AAAAAAAAAA | | PP | | DD | DD | EE | | FF | |
| DD | DD | AAAAAAAAAA | | PP | | DD | DD | EE | | FF | |
| DD | DD | AA | AA | PP | | DD | DD | EE | | FF | |
| DD | DD | AA | AA | PP | | DD | DD | EE | | FF | |
| DD | DD | AA | AA | PP | | DD | DD | EE | | FF | |
| DDDDDDDD | | AA | AA | PP | | DDDDDDDD | | EEEEEEEE | | FF | |
| DDDDDDDD | | AA | AA | PP | | DDDDDDDD | | EEEEEEEE | | FF | |

....
....
....
....

| | | | | | | | |
|-----|-----|----------|----|------------|--|----|--|
| MM | | MM | | DDDDDDDD | | LL | |
| MM | | MM | | DDDDDDDD | | LL | |
| MMM | MMM | DD | DD | LL | | LL | |
| MMM | MMM | DD | DD | LL | | LL | |
| MM | MM | DD | DD | LL | | LL | |
| MM | MM | DD | DD | LL | | LL | |
| MM | MM | DD | DD | LL | | LL | |
| MM | MM | DD | DD | LL | | LL | |
| MM | MM | DD | DD | LL | | LL | |
| MM | MM | DD | DD | LL | | LL | |
| MM | MM | DD | DD | LL | | LL | |
| MM | MM | DDDDDDDD | | LLLLLLLLLL | | | |
| MM | MM | DDDDDDDD | | LLLLLLLLLL | | | |

.TITLE \$DAPDEF - DATA ACCESS PROTOCOL DEFINITIONS
.IDENT 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

++
Facility: DAP (Data Access Protocol)

Abstract:

This module defines the DAP control block. It is both an input and
output control structure for the FAL\$DECODE_MSG and NT\$DECODE_MSG
subroutines in FAL and RMS, respectively.

Environment:

The MDL translator must be used to convert DAPDEF.MDL into
DAPDEF.MAR (and DAPDEF.B32).

Author: James A. Krycka, Creation Date: 17-OCT-1977

Modified By:

V03-007 JEJ0018 J E Johnson 27-Mar-1984
Correct double assignment of DAP\$V_POS caused in V03-006; now
P/OS will be identified as DAP\$V_P_OS and DAP\$K_P_OS.
Remove no longer used DAP buffer size constants:
DAP\$K_INIBUFSIZ, DAP\$K_MINBUFSIZ, and DAP\$K_MAXBUFSIZ.

V03-006 JAK0124 J A Krycka 06-SEP-1983
Define operating system class bits analogous to DAP\$V_VAXVMS
(VAXELAN, TOPS10, TOPS20, RT11, RSTS, RSX, IAS, and POS).
Define DAP\$B_X_FIELD containing flags from DAP\$Q_DCODE_FLG.

Rearrange order of DAP\$Q_DCODE_FLG bits.

- V03-005 JAK0112 J A Krycka 22-JUN-1983
Define DAP\$V_GEQ_V71.
Define DAP\$V_VMS_XPF1 thru VMS_XPF4.
- V03-004 JAK0111 J A Krycka 17-JUN-1983
Upgrade definitions to correspond to DAP V7.0 specification:
Define DAP\$K_VAXELAN and DAP\$K_RMS32S.
Define new SYSCAP bit (OCTALVER).
Define DAP\$K_IN8 and DAP\$K_BN8.
- V03-003 KRM0102 K Malik 09-May-1983
Define new SYSCAP field bits (MODATTCRE, NAM3PART, CHGATTREN,
CHGTIMREN, CHGPROREN, BLKCNT).
Rename SYSCAP bits (CHGATT to CHGATTCLS, CHGTIM to CHGTIMCLS,
CHGPRO to CHGPROCLS, CHGNAM to CHGNAMCLS).
Define DAP\$V_BLKCNT, DAP\$B_BLKCNT, and DAP\$C_BLKCNT.
Define DAP\$V_DSP_3NAM.
Define DAP\$K_QUIT.
- V03-002 KRM0081 K Malik 23-Mar-1983
Define DAP\$V_GEQ_V70.
Rename DAP\$B_SOFTVER to DAP\$B_DECVER.
Rename DAP\$B_USRISOFT to DAP\$K_USRVER.
Define DAP\$K_STMLF and DAP\$K_STMCR.
- V03-001 KRM0065 K Malik 23-NOV-1982
Change DAP\$K_SYSCAP2_V and DAP\$K_VALID_R2F values to support
rename operation.
- V02-047 JAK0070 J A Krycka 27-JAN-1982
Remove all "DAP\$V_..." symbols from expressions and eliminate
the use of "." in symbol names to aid in future conversion of
this MDL file into SDL format.
- V02-046 JAK0063 J A Krycka 24-AUG-1981
Cleanup:
Rearrange sections defined by \$DAPPLGDEF.
Expand several menu fields from one byte to two bytes in length
(DAP\$W_CTLMENU, DAP\$W_TIMENU, DAP\$W_PROMENU, DAP\$W_SUMENU).
For consistency, denote fields that exist in two messages as
DAP\$s_name1 and DAP\$s_name2 (FOP, ALQ, DEQ, DISPLAY, RECNUM).
Remove unused system specific fields (DAP\$L_FOP, DAP\$L_ROP, and
DAP\$L_CTX).
Rename SYSCAP bits (RANREC to RANRRN, MULKEY to IDXORG, and
BITCOUNT to BITOPT).
- V02-045 JAK0063 J A Krycka 21-AUG-1981
Upgrade definitions to correspond to DAP V6.0 specification:
Define DAP\$V_GEQ_V60.
Define DAP\$V_EXTEND and DAP\$V_DISPLAY.
Define new SYSCAP field bits (TEXTEND, DISPLAY, GNGOPT, CHGATT,
CHGTIM, CHGPRO, and CHGNAM).
Define new FOP field bit (DIR).
Define new ROP field bits (ROPWAT, RRL, and REA).

Define DAP\$K_EXTEND_B and DAP\$K_EXTEND_E; remove DAP\$K_EXTEND.
Define DAP\$K_CHANGE_B, DAP\$K_CHANGE_E, and DAP\$K_TERMINATE.
Rename DAP\$K_PURGE to DAP\$K_RESET.
Define DAP\$Q_STX and DAP\$ STX.
Define DAP\$V_PDT, DAP\$Q_PDT, and DAP\$ PDT.
Define DAP\$V_ADT, DAP\$Q_ADT, and DAP\$ ADT.
Modify value of DAP\$K_SYSCAP1_V (set EXTEND and DISPLAY bits).
Modify value of DAP\$K_SYSCAP2_V (set CHGTIM and CHGPRO bits).

V02-044 JAK0061 J A Krycka 17-JUL-1981
Define DAP\$K_INIBUFSIZ, DAP\$K_MINBUFSIZ, and DAP\$K_MAXBUFSIZ.
Remove DAP\$K_BUFSIZ_F and DAP\$K_BUFSIZ_R.

V02-043 JAK0060 J A Krycka 23-JUN-1981
Define DAP\$K_TOPS10, and DAP\$K_TOPS10F.
Define DAP\$V_BDT, DAP\$Q_BDT, and DAP\$ BDT.
Modify value of DAP\$K_FLAGS_U (remove LEN256 bit).
Modify value of DAP\$K_SYSCAP1_V (set RANRFA and BIGBLK bits).

V02-042 JAK0050 J A Krycka 22-NOV-1980
Define DAP\$V_RMS and DAP\$V_FCS.
Fix bug in definition of reserved bit in FOP field.
Change DAP\$K_BUFSIZ_F value from <4096+256> to <4096+32>.
Modify value of DAP\$K_SYSCAP2_V (include WILDCARD bit).

V02-041 REFORMAT J A Krycka 26-JUL-1980

```

:++
: Define the overall structure of the DAP control block and symbols related
: to its prologue section.

```

```

: Note: Longword and quadword fields are longword aligned within the control
: block. Fields longer than 8 bytes are not stored within. Instead a
: descriptor is stored in the control block that points to an external
: buffer where the field data is located.
:--

```

```

$STRUCT DAP,PLGDEF      : DAP control block prologue
                          : -----
                          : Parameter and status section
                          : -----
F DCODE_FLG,Q           : Message decode status flags
                          : (output from message decode subroutine)
                          : Note: bits 00-31 are defined external to DAP
                          : Note: bits 32-63 are defined by DAP herein
                          : Remote DAP protocol version level (bits 32-47)
                          : Remote system classification (bits 48-63)
                          : Meaning:
S VERSION,4,W           : Skip over reserved bits
S PARTNER,6,W           : Partner implemented to DAP since V4.1
V <                     : Partner implemented to DAP since V4.2
  ,32                   : Partner implemented to DAP since V5.2
  GEQ_V41               : Partner implemented to DAP since V5.4
  GEQ_V42               : Partner implemented to DAP since V5.6
  GEQ_V52               : Partner implemented to DAP since V6.0
  GEQ_V54               : Partner implemented to DAP since V7.0
  GEQ_V56               : Partner implemented to DAP since V7.1
  GEQ_V60               : Spare
  GEQ_V70               : VAX/VMS experimental protocol option flag
  GEQ_V71               : VAX/VMS experimental protocol option flag
  ,4                    : VAX/VMS experimental protocol option flag
  VMS_XPF1              : VAX/VMS experimental protocol option flag
  VMS_XPF2              : VAX/VMS experimental protocol option flag
  VMS_XPF3              : VAX/VMS experimental protocol option flag
  VMS_XPF4              : VAX/VMS experimental protocol option flag
  RMS                  : Partner uses an RMS based file system
  FCS                  : Partner uses an FCS based file system
  STM_ONLY              : Partner uses a stream ASCII based file system
  ,1                    : Spare
  VAXVMS               : Partner runs under VAX/VMS
  VAXELAN              : Partner runs under VAXELAN
  TOPS10               : Partner runs under TOPS-10
  TOPS20               : Partner runs under TOPS-20
  RT11                 : Partner runs under RT-11
  RSTS                 : Partner runs under RSTS/E
  RSX                  : Partner runs under RSX-11M, -11MP, or -11S
  IAS                  : Partner runs under IAS or RSX-11D
  P_OS                 : Partner runs under PO/S
  ,3                    : Spare
  >
F MSG_BUF1,Q           : On input, descriptor of message string
                          : to decode
                          : On output, descriptor of string remaining
                          : after message just decoded
F MSG_BUF2,Q           : On input, ignored

```

F DCODE_STS,L

S ,0,B

S DCODE_FID,1,B

S DCODE_MSG,2,B

S DCODE_MAC,3,B

F MSG_MASR,L

F CRC_RSLT,L

F X_FIELD,B

V<

X_RECNUM

X_CHECK

,6

>

F ,B,3

F ,L,2

F CMWA,L,20

K CMWA,<20*4>

S ,0,L,4

S ,4,L,16

F SSPWA,L,4

K SSPWA,<4*4>

S ,0,L,4

F TEMP,L,4

K TEMP,<4*4>

F ,L,8

L BLN

E

```

: On output, descriptor of message just
: decoded; same as MSG_BUG1 on input if
: no blocked message follows
: Message decode status codes
: (output from message decode subroutine)
: Message decode success/fail (1/0) status flag
: On error, DAP field ID code; else 0
: Message type (0 if invalid)
: On error, DAP MACCODE error code; else 0
: Bit mask of valid messages to receive
: (input to message decode subroutine)
: (bit offsets are derived from message type
: values, e.g., offset for Data message is
: <10DAP$K DAT_MSG>)
: Current CRC resultant value
: Explicit field found in message flags field
: Meaning:
: Message explicitly contained RECNUM field
: Message explicitly contained CHECK field
: Spare
:
: -----
: Message decode section (part 1)
: -----
: Configuration message save section
: (space for DAP$Q_SYSCAP bit mask field
: defined by the $DAPCNFDEF macro)
: -----
: Message decode section (parts 2 and 3)
: -----
: Current message work area
: Current message work area size
: Message header section
: (space for current message header fields
: defined by the $DAPHDRDEF macro)
: Message operand section
: (space for current message operand fields
: defined by the $DAPxxxDEF macros, where xxx
: represents the 15 DAP message mnemonics)
: ***** offset = ^X80 = 128 *****
: -----
: Message decode section (parts 4 and 5)
: -----
: System specific work area
: System specific work area size
: System specific section
: (space for system specific fields
: defined by the $DAPSSPDEF macro)
: Temporary work area
: Temporary work area size
: Spare
: -----
: Define length of DAP control block
:

```

```

:++
: Define symbols related to the DAP message header.
:--

```

```

$STRUCT DAP,HDRDEF      : DAP message header
F .L,12                  : Position to message header section
                           : of DAP control block
F TYPE,B                 : DAP message type field (1) : B
K <                       : DAP message type:
  CNF_MSG,1               : Configuration message
  ATT_MSG,2               : Attributes message
  ACC_MSG,3               : Access message
  CTL_MSG,4               : Control message
  CON_MSG,5               : Continue Transfer message
  ACK_MSG,6               : Acknowledge message
  CMP_MSG,7               : Access Complete message
  DAT_MSG,8               : Data message
  STS_MSG,9               : Status message
  KEY_MSG,10              : Key Definition Attributes message
  ALL_MSG,11              : Allocation Attributes message
  SUM_MSG,12              : Summary Attributes message
  TIM_MSG,13              : Date and Time Attributes message
  PRO_MSG,14              : Protection Attributes message
  NAM_MSG,15              : Name message
                           : (16) reserved for ACL Attributes message
  >
K VALID_R2F,-            : Mask of DAP messages valid for RMS to send:
  <^X0000EDBE>           : CNF, ATT, ACC, CTL, CON, CMP, DAT, KEY, ALL,
                           : TIM, PRO, NAM
K VALID_F2R,-            : Mask of DAP messages valid for FAL to send:
  <^X0000FFC6>           : CNF, ATT, ACK, CMP, DAT, STS, KEY, ALL, SUM,
                           : TIM, PRO, NAM
F FLAGS,B                : DAP message flags field (EX-5) : BM
V <M                      : Menu of fields to follow:
  STREAMID                : STREAMID
  LENGTH                  : LENGTH
  LEN256                  : LEN256
  BITCNT                  : BITCNT
  TMP1$,1                 : Reserved
  SYSPEC                  : SYSPEC
  SEGMENT                  : Flags field options:
                           : This is a segmented DAP message with
                           : at least one more segment to follow
  TMP2$,1                 : Reserved
  >
K FLAGS_I,<-              : Define flags options that are invalid:
  <DAP$M_TMP1$>!--       : Reserved
  <DAP$M_TMP2$>!--       : Reserved
  0>
K FLAGS_U,<-              : Define flags options unsupported by VAX:
  <DAP$M_BITCNT>!--      : BITCNT
  <DAP$M_SEGMENT>!--     : SEGMENT
  0>
F STREAMID,B              : Data stream identification field (1) : B
F LENGTH,B                : Length (of rest of message) field (1) : B

```

F LEN256,B
F BITCNT,B
F B,2
F SYSPEC,Q
E

: Length extension field (1) : B
: Bit count field (1) : B
: Padding
: Descriptor pointing to the
: System specific field (1-255) : B
:

```

:++
: Define symbols related to the system specific field (mini-message)
: contained in the DAP message header.
:--

```

```

$STRUCT DAP,SSPDEF      : System specific field
F ,L,32                  : Position to system specific section
                          : of DAP control block
F SSP_MENU,W             : System specific menu field (EX-5) : B
  V <M                   : Menu of fields to follow:
    SSP_CAP               : Extended system capabilities
    SSP_FLG               : Extended flags
    TMP1$,14              : Reserved
  >
  K SSP_MENU_I,<-         : Define SSP_MENU options that are invalid:
    <DAPSM_TMP1$>!--     : Reserved
    0>
  K SSP_MENU_U,<-         : Define SSP_MENU options unsupported by VAX:
    0>
F ,W                     : Padding
F SSP_FLG,L              : System specific flags field (EX-5) : B
  V <M                   : Meaning:
    LOAD                  : Load image modifier for open function
    TMP1$,31              : Reserved
  >
  K SSP_FLG_I,<-          : Define SSP_FLG options that are invalid:
    <DAPSM_TMP1$>!--     : Reserved
    0>
  K SSP_FLG_U,<-          : Define SSP_FLG options unsupported by VAX:
    0>
F SSP_CAP,L              : System specific capabilities field (EX-5) : B
  V <M                   : Partner node supports:
    LOADIM                : Load image function
    ,31                   : Reserved
  >
  K SSP_CAP_V,<-          : Define SSP_CAP options supported by VAX:
    <DAPSM_LOADIM>!--    : LOADIM
    0>
F ,L,1                   : Spare
E

```

```

**
Define symbols related to the Configuration message (TYPE=1).
--

```

```

$STRUCT DAP,CNFDEF      : DAP Configuration message
M 1                      :
F ,L.16                  : Position to message operand section
                          : of DAP control block
F BUFSIZ,W               : Buffer size field (2) : B
                          : (This is DAP buffer size value from partner)
F OSTYPE,B               : Operating system type field (1) : B
K <                      : Operating system type:
  RT11,1                 : RT-11
  RSTS,2                 : RSTS/E
  RSX11S,3               : RSX-11S
  RSX11M,4               : RSX-11M
  RSX11D,5               : RSX-11D
  IAS,6                  : IAS
  VAXVMS,7               : VAX/VMS
  TOPS20,8               : TOPS-20
  TOPS10,9               : TOPS-10
  RTS8,10                : RTS-8
  OS8,11                 : OS-8
  RSX11MP,12             : RSX-11M-PLUS
  COPOS11,13             : TOPS-20 (using 2050/2060 front end)
  P OS,14                : P/OS
  VAXELAN,15             : VAXELAN
>
F FILESYS,B              : File system type field (1) : B
K <                      : File system type:
  RMS11,1                : RMS-11
  RMS20,2                : RMS-20
  RMS32,3                : RMS-32
  FCS11,4                : FCS-11
  RT11FS,5               : RT-11
  NO FS,6                : No file system present
  TOPS20FS,7             : TOPS-20
  TOPS10FS,8             : TOPS-10
  OS8FS,9                : OS-8
  RMS32S,10              : RMS-32 subset
>
F VERNUM,B               : DAP version number field (1) : B
K VERNUM_V,7             : Value for VAX/VMS V4.0
F ECONUM,B               : ECO version number field (1) : B
K ECONUM_V,0             : Value for VAX/VMS V4.0
F USRNUM,B               : User protocol version number field (1) : B
K USRNUM_V,0             : Value for VAX/VMS V4.0
F DECVER,B               : DEC software version number field (1) : B
K DECVER_V,4             : Value for VAX/VMS V4.0
F USRVER,B               : User software version number field (1) : B
K USRVER_V,0             : Value for VAX/VMS V4.0
F ,B,3                   : Padding
F ,L,13                  : Spare
P 1
F ,L,10                  : Position to Configuration message save section

```

```

F SYSCAP,Q
V <
  FILALL
  SEQORG
  RELORG
  .1
  EXTEND
  SEQFIL
  RANRRN
  RANVBN
  RANKEY
  .1
  RANRFA
  IDXORG
  SWMODE
  APPEND
  SUBMIT
  .1
  MDS
  DISPLAY
  MSGBLK

  UNRBLK
  BIGBLK

  DAPCRC
  KEYXAB
  ALLXAB
  SUMXAB
  DIRECTORY
  TIMXAB
  PROXAB
  .1
  FOPSPL
  FOPSCF
  FOPDLT
  >
V <
  .32
  .1
  SEQRAC
  .1
  BITOPT
  WARNING

  RENAME
  WILDCARD
  GNGOPT
  NAMMSG
  SEGMSG
  CHGATTCLS
  CHGTIMCLS
  CHGPROCLS
  CHGNAMCLS

  MODATTCRE

```

```

: System capabilities field (EX-12) : BM
: Partner node supports:
: Allocation of space at file creation
: Sequential file organization
: Relative file organization
: Reserved for HSHORG
: Manual file extension
: Sequential file access (file transfer mode)
: Random access by relative record number
: Random access by virtual block number
: Random access by key value
: Reserved for RANHSH
: Random access by record file address
: Multi-keyed indexed file organization
: Dynamic switching of access modes
: Append records to end-of-file
: Command file submission/execution
: Reserved for COMPRESS (data compression)
: Multiple data streams per file
: Display of file attributes on request
: Blocking of DAP messages up to response
:   using a 1-byte length field (LENGTH)
: Unrestricted blocking of DAP messages
: Blocking of DAP messages up to response
:   using a 2-byte length field (LEN256,LENGTH)
: DAP message CRC checksum
: Key Definition XAB message
: Allocation XAB message
: Summary XAB message
: Directory list operation
: Date and Time XAB message
: File Protection XAB message
: Reserved for ACLXAB
: Spool file on close FOP option
: Execute command file on close FOP option
: Delete file on close FOP option

: Partner node supports:
: (skip over bits defined above)
: Reserved for DFTFIL (default file spec)
: Sequential record access
: Reserved for RECOVERY
: Bit count option in the FLAGS field
: Warning Status message and associated error
:   recovery message exchange
: File rename operation
: Wildcard operations (excluding directory)
: Go/Nogo option in the ACCOPT field
: Name message
: Segmented DAP messages
: Changing file attributes on close via ATT msg
: Changing file attributes on close via TIM msg
: Changing file attributes on close via PRO msg
: Changing file attributes on close via NAM msg
:   (i.e., rename of file)
: Modified attributes returned on create

```

```
NAM3PART      : 3-part Name message format in DISPLAY field
                of both Access and Control messages
CHGATTREN      : Changing file attributes on rename via ATT msg
CHGTIMREN      : Changing file attributes on rename via TIM msg
CHGPROREN      : Changing file attributes on rename via PRO msg
BLKCNT         : BLKCNT field in Control message
OCTALVER       : Octal version numbers only in file specs
                (bit is valid only for DAP V7.0 or later)
                Reserved
,11
>
K SYSCAP1 V,-  : Define supported SYSCAP options (bits 00-31):
<^XEFF67DF7>   FILALL, SEQORG, RELORG, EXTEND, SEQFIL,
                RANRRN, RANVBN, RANKEY, RANRFA, IDXORG, SWMODE,
                APPEND, SUBMIT, DISPLAY, MSGBLK, BIGBLK,
                DAPCRC, KEYXAB, ALLXAB, SUMXAB, DIRECTORY,
                TIMXAB, PROXAB, FOPSPL, FOPSCF, FOPDLT
K SYSCAP2 V,-  : Define supported SYSCAP options (bits 32-63):
<^X0000T962>   SEQRAC, RENAME, WILDCARD, NAMMSG, CHGTIMCLS,
                CHGPROCLS
E
```

```

:++
: Define symbols related to the Attributes message (TYPE=2).
:--

```

```

$STRUCT DAP,ATTDEF      : DAP Attributes message
F .L,16                  : Position to message operand section
                           : of DAP control block
F ATTMENU,L              : Attributes menu field (EX-6) : BM
V <M                     : Menu of fields to follow:
  DATATYPE                : Data type
  DRG                     : File organization
  RFM                     : Record format
  RAT                     : Record attributes
  BLS                     : Block size
  MRS                     : Maximum record size
  ALQ1                    : Allocation quantity
  BKS                     : Bucket size
  FSZ                     : Fixed control area size
  MRN                     : Maximum record number
  RUNSYS                  : Run-time system identification
  DEQ1                    : Default extension quantity
  FOP1                    : File options
  BSZ                     : Byte size field
  DEV                     : Device characteristics
  TMP1$,1                 : Reserved for SDC
  LRL                     : Longest record length
  HBK                     : Highest virtual block number
  EBK                     : End-of-file block number
  FFB                     : First free byte in end-of-file block
  SBN                     : Starting logical block number
  TMP2$,11                : Reserved
>
K ATTMENU_I,<-            : Define ATTMENU options that are invalid:
  <DAPSM_TMP1$>!--       : Reserved
  <DAPSM_TMP2$>!--       : Reserved
  0>
K ATTMENU_U,<-            : Define ATTMENU options unsupported by VAX:
  0>
F DATATYPE,B             : Data type field (EX-2) : BM
V <M                     : Define offsets and masks:
  ASCII                  : Data in ASCII format
  IMAGE                  : Data in IMAGE format
  TMP1$,1                : Reserved for EBCDIC
  CMPFMT                 : Compressed format
  EXEC                   : File contains executable code
  PRIV                   : File contains privileged code
  TMP2$,1                : Reserved (ignore if received)
                           : (this was attributes match flag in DAP V4.1)
  ZERO                   : Zero file on erase file operation
>
K DATATYP_I,<-            : Define DATATYPE options that are invalid:
  <DAPSM_TMP1$>!--       : Reserved
  0>
K DATATYP_U,<-            : Define DATATYPE options unsupported by VAX:
  <DAPSM_CMPFMT>!--      : CMPFMT

```

```

      <DAPSM_ZERO>!--
      0>
      K DATATYP D,<--
      <DAPSM_IMAGE>!--
      0>
F   ORG,B
      K <
      SEQ,0
      REL,16
      IDX,32
      >
      K ORG_D,DAPSK_SEQ
F   RFM,B
      K <
      UDF,0
      FIX,1
      VAR,2
      VFC,3
      STM,4
      STMLF,5
      STMCR,6
      >
      K RFM_D,DAPSK_FIX
F   RAT,B
      V <M
      FTN
      CR
      PRN
      BLK
      EMBEDDED
      TMP1$,1
      LSA
      MACY11
      >
      K RAT_I,<--
      <DAPSM_TMP1$>!--
      0>
      K RAT_U,<--
      <DAPSM_LSA>!--
      <DAPSM_MACY11>!--
      0>
      K RAT_D,<--
      <DAPSM_EMBEDDED>!--
      0>

F   BLS,W
      K BLS_D,512
F   MRS,W
F   ALQ1,L
F   BKS,B
F   FSZ,B
F   BSZ,B
      K BSZ_D,8

```

```

      ZERO
      Define default DATATYPE value
      IMAGE
      File organization field (1) : B
      File organization:
      Sequential
      Relative
      Indexed
      (48) reserved for hash
      Define default ORG value
      Record format field (1) : B
      Record format:
      Undefined
      Fixed length
      Variable length
      Variable length with fixed control
      Stream ASCII
      Stream LF
      Stream CR
      Define default RFM value
      Record attributes field (EX-3) : BM
      Meaning:
      Fortran carriage control
      Implied (LF-Record-CR) carriage control
      Print file format
      Records do not cross block boundaries
      Records have embedded control characters
      Reserved
      Line sequenced ASCII
      MACY11 format
      Define RAT options that are invalid:
      Reserved
      Define RAT options unsupported by VAX:
      LSA
      MACY11
      Define default RAT value
      EMBEDDED
      ***** No default value is stated in the
      ***** DAP spec although some systems
      ***** treat EMBEDDED as the default
      Block size field (2) : B
      Define default BLS value
      Maximum record size field (2) : B
      Allocation quantity field (1-5) : B
      Bucket size field (1) : B
      Fixed control area size field (1) : B
      Byte size field (1) : B
      Define default BSZ value

```

```

F ,B
F DEQ1,W
F ,B,2
F MRN,L
F RUNSYS,Q
F FOP1,L
V <M
  RWO
  RWC
  TMP1$,1
  POS
  DLK
  DIR
  FLK
  CTG
  SUP
  NEF
  TMP
  TMD
  TMP2$,1
  DMO
  WCK
  RCK
  CIF
  TMP3$,1
  SQO
  MXV
  SPL
  SCF
  DLT

  CBT
  TMP4$,1
  DFW
  TEF
  OFP
  TMP5$,4
  >
K FOP_1,<-
  <DAPSM_TMP1$>!--
  <DAPSM_TMP2$>!--
  <DAPSM_TMP3$>!--
  <DAPSM_TMP4$>!--
  <DAPSM_TMP5$>!--
  0>
K FOP_U,<-
  <DAPSM_DMO>!--
  0>
F DEV,L
V <
  DEVREC
  DEVCLL

```

Padding
 Default extension quantity field (2) : B
 Padding
 Maximum record number field (1-5) : B
 Descriptor pointing to the
 Run-time system field (1-40) : A
 File options field (EX-6) : BM
 Options:
 Rewind magtape on open
 Rewind magtape on close
 Reserved
 Position magtape past last created file
 Do not lock file if improperly closed
 Directory file
 File locked
 Contiguous space allocation
 Supersede existing file on create
 Inhibit positioning magtape to end-of-file
 Create temporary file
 Create temporary file and mark for delete
 Reserved
 Dismount magtape on close
 Enable write checking
 Enable read checking
 Create if no file present else open file
 Reserved for LKO
 Sequential access only
 Maximize version number
 Spool file on close
 Submit command file on close
 Delete file on close
 (used stand-alone or as a suboption to
 SCF or SPL)
 Contiguous-best-try space allocation
 Reserved for WAT
 Deferred write (REL and IDX files)
 Truncate at EOF on close (SEQ files)
 Output file parse
 Reserved
 Define FOP options that are invalid:
 (This is used for both FOP1 and FOP2 fields)
 Reserved
 Reserved
 Reserved
 Reserved
 Reserved
 Define FOP options unsupported by VAX:
 (This is used for both FOP1 and FOP2 fields)
 DMO
 Note: allow DLK, DIR, and FLK
 Device characteristics field (EX-6) : BM
 Meaning:
 Device is record oriented
 Carriage control device

```

DEVTRM      : Device is a terminal
DEVDIR      : Device is directory structured
DEVSDI      : Device is single directory structured
DEVSQD      : Seq. block oriented device (e.g., magtape)
TMP1$,1,,M  : Reserved
DEVFOD      : Files oriented device (e.g., disk, magtape)
DEVSHR      : Device is sharable
DEVSP      : Device is being spooled
DEVMNT      : Device is mounted
DEVDMT      : Device is marked for dismount
DEVALL      : Device is allocated
DEVIDV      : Device is capable of providing input
DEVODV      : Device is capable of providing output
DEVSWL      : Device is software write locked
DEVAVL      : Device is available
DEVELG      : Device has error logging enabled
DEVMBX      : Device is a mailbox
DEVRTM      : Device is realtime in nature
DEVRND      : Device allows random access
DEVRCK      : Device has read checking enabled
DEVWCK      : Device has write checking enabled
DEVFOR      : Device is mounted as foreign (not files-11)
DEVNET      : Network device
DEVGEN      : Generic device
TMP2$,6,,M  : Reserved
>
K DEV_I,<-   : Define DEV options that are invalid:
<DAPSM_TMP1$>!-- Reserved
<DAPSM_TMP2$>!-- Reserved
0>
K DEV_U,<-   : Define DEV options unsupported by VAX:
0>
F ,L,1       : Reserved for SDC
F LRL,W      : Longest record length field (2) : B
F FFB,W      : First free byte in EOF block field (2) : B
F HBK,L      : Highest virtual block number field (1-5) : B
F EBK,L      : End-of-file block number field (1-5) : B
F SBN,L      : Starting logical block number field (1-5) : B
E

```



```

SHRGET
SHRDEL
SHRUPD
MSE
UPI
NIL
TMP18,1
>
K SHR_I,<-
  <DAPSM_TMP18>!--
  0>
K SHR_U,<-
  <DAPSM_MSE>!--
  0>
K SHR_D,<-
  0>
F FILESPEC,0
F DISPLAY1,W
V <M
  DSP_ATT
  DSP_KEY
  DSP_ALL
  DSP_SUM
  DSP_TIM
  DSP_PRO
  TMP18,2
  DSP_NAM
  DSP_3NAM
  TMP28,6
  >
K DISPLAY_I,<-
  <DAPSM_TMP18>!--
  <DAPSM_TMP28>!--
  0>
K DISPLAY_U,<-
  <DAPSM_DSP_3NAM>!--
  0>
F W
F PASSWORD,0
F .L,10
E

```

```

: Get record
: Delete record
: Update record
: Multiple record streams enabled
: User provided interlocking
: No shared access allowed
: Reserved
:
: Define SHR options that are invalid:
:   Reserved
:
: Define SHR options unsupported by VAX:
:   MSE
:
: Define default SHR value
: ***** This is contrary to the DAP spec
: ***** which says that DAPSM_GET is the default
: Descriptor pointing to the
: File specification field (I-255) : A
: Display attributes field (EX-4) : BM
: Return the following:
:   Attributes message
:   Key Definition Attributes message
:   Allocation Attributes message
:   Summary Attributes message
:   Date and Time Attributes message
:   Protection Attributes message
:   Reserved
:   Reserved for ACL Attributes message
:   Name message
:   3-part Name message
:   Reserved
:
: Define DISPLAY options that are invalid:
:   (This is used for both DISPLAY1 and DISPLAY2)
:   Reserved
:   Reserved
:
: Define DISPLAY options unsupported by VAX:
:   (This is used for both DISPLAY1 and DISPLAY2)
:   3-Part Name message
:
: Padding
: Descriptor pointing to the
: Password field (I-40) : B
: Spare
:

```

```

:++
: Define symbols related to the Control message (TYPE=4).
:--

```

```

$STRUCT DAP,CTLDEF      : DAP Control message
F .L,16                  : Position to message operand section
                           of DAP control block
F CTLFUNC,B              : Control function field (1) : B
K <                      : Control function:
  GET_READ,1             : Get record or read block
  CONNECT,2              : Establish data stream
  UPDATE,3               : Update record
  PUT_WRITE,4            : Put record or write block
  DELETE,5               : Delete record
  REWIND,6               : Rewind file
  TRUNCATE,7             : Truncate sequential file
                           (8) reserved for modify file attributes
  RELEASE,9              : Release locked record
  FREE,10                : Free all locked records
  EXTEND_B,11            : Extend file (beginning message of sequence)
  FLUSH,12               : Flush all records
                           (13) reserved for next volume processing
  FIND,14                : Find record
  EXTEND_E,15            : Extend file (ending message of sequence)
  DISPLAY,16             : Display file attributes
  SPACE_FW,17            : Space file forward
  SPACE_BW,18            : Space file backward
                           (19) reserved for checkpoint file
                           (20) reserved for recovery get
                           (21) reserved for recovery put
>
F .B,3                   : Padding
F CTLMENU,W              : Control menu field (EX-4) : BM
V <M                     : Menu of fields to follow:
  RAC                    : RAC
  KEY                    : KEY
  KRF                    : KRF
  ROP                    : ROP
  TMP1$,1               : Reserved for HSH
  DISPLAY2               : DISPLAY2
  BLKCNT                 : BLKCNT
  TMP2$,9               : Reserved
>
K CTLMENU I,<-           : Define CTLMENU options that are invalid:
  <DAP$M_TMP1$>!-       : Reserved
  <DAP$M_TMP2$>!-       : Reserved
  0>
K CTLMENU U,<-           : Define CTLMENU options unsupported by VAX:
  <DAP$M_BLKCNT>!-      : BLKCNT
  0>
F RAC,B                  : Record access field (1) : B
K <                      : Record access type:
  SEQ_ACC,0             : Sequential record access
  KEY_ACC,1             : Random access by key value or record number
  RFA_ACC,2             : Random access by RFA

```

| | |
|-----------------------|--|
| SEQ_FILE,3 | Sequential file transfer mode |
| BLK_VBN,4 | Block I/O access by VBN |
| BLK_FILE,5 | Block I/O file transfer mode |
| > | |
| K RAC_D,DAPSK_SEQ_ACC | Define default RAC value |
| F KRF,B | Key of reference field (1) : B |
| F KEY,Q | Descriptor pointing to the |
| | Key field (1-255) : B |
| F ROP,L | Record options field (EX-6) : BM |
| V <M | Meaning: |
| EOF | Position to end-of-file |
| FDL | Fast record delete |
| UIF | Convert put to update function as required |
| TMP1\$,1 | Reserved for HSM |
| LOA | Load buckets according to bucket fill size |
| ULK | Enable manual unlocking of records; |
| | disable automatic unlocking |
| TPT | Truncate put; write EOF then put (SEQ files) |
| RAH | Read ahead |
| WBH | Write behind |
| KGE | Key value is greater than or equal |
| KGT | Key value is greater than |
| NLK | Do not lock record |
| RLK | Read of locked record allowed |
| ROPB10 | Connect for block I/O operations only |
| LIM | Compare for key limit reached |
| NXR | Non-existent record processing |
| ROPWAT | Wait until locked record becomes available |
| RRL | Read record regardless of lock |
| REA | Lock record but allow others to read it |
| TMP2\$,13 | Reserved |
| > | |
| K ROP_I,<- | Define ROP options that are invalid: |
| <DAPSM_TMP1\$>!-- | Reserved |
| <DAPSM_TMP2\$>!-- | Reserved |
| 0> | |
| K ROP_U,<- | Define ROP options unsupported by VAX: |
| 0> | |
| F DISPLAY2,W | Display attributes field (EX-4) : BM |
| | (see DISPLAY1 field of Access message |
| | for bit definitions) |
| F BLKCNT,B | Block count field |
| F ,B | Padding |
| F ,L,10 | Spare |
| E | |

++
: Define symbols related to the Continue Transfer message (TYPE=5).
--

```
$STRUCT DAP,CONDEF      : DAP Continue Transfer message
F .L,16                 : Position to message operand section
                          : of DAP control block
F CONFUNC,B             : Continue transfer function field (1) : B
  K <                   : Recovery action:
    RETRY,1              : Try access function again
    SKIP_REC,2           : Skip record in error and continue
    ABORT,3              : Abort request
    RESUME,4             : Resume operation
    QUIT,5               : Terminate file processing
  >
F .B,3                  : Padding
F .L,15                 : Spare
E
```

++
: Define symbols related to the Acknowledge message (TYPE=6).
--

| | | |
|---------------------|---|-------------------------------------|
| \$STRUCT DAP,ACKDEF | : | DAP Acknowledge message |
| F ,L,16 | : | Position to message operand section |
| F ,L,16 | : | of DAP control block |
| E | : | Spare |
| | : | |

```

:++
: Define symbols related to the Access Complete message (TYPE=7).
:--

```

```

$STRUCT DAP,CMPDEF      : DAP Access Complete message
F .L,16                 : Position to message operand section
                          : of DAP control block
F CMPFUNC,B             : Access complete function field (1) : B
K <                     : Access complete function:
  CLOSE,1               : Close file
  RESPONSE,2            : Response to partner's CMPFUNC request
  RESET,3               : Close file and restore it to initial state
                          : (this used to be named PURGE)
  DISCONN,4             : Disconnect record stream
  SKIP_FILE,5           : Skip to next file (i.e., close this file
                          : and open next file)
  CHANGE_B,6            : Close file and change its file attributes
                          : (beginning message of sequence)
  CHANGE_E,7            : Close file and change its file attributes
                          : (ending message of sequence)
  TERMINATE,8           : Terminate (abort) operation and re-initialize
  >
F .B                    : Padding
F CHECK,W               : CRC Checksum field (2) : B
F FOP2,L                : File options field (EX-6) : BM
                          : (see FOP1 field of Attributes message
                          : for bit definitions)
F .L,14                 : Spare
E

```



```

:++
: Define symbols related to the Status message (TYPE=9).
:--

```

| | |
|----------------------|--|
| \$STRUCT DAP,ST\$DEF | DAP Status message |
| F .L,16 | Position to message operand section of DAP control block |
| F ST\$CODE,W | DAP status code field (2) : B |
| V <M | Subfields: |
| MICCODE,12 | Micro status code |
| MACCODE,4 | Macro status code |
| > | |
| K <,S | MACCODE field status code classes: |
| PENDING,0 | Operation in progress |
| SUCCESS,1 | Operation completed successfully |
| UNSUPPORT,2 | DAP implementation does not support request (3) reserved |
| FILE_OPEN,4 | Error related to opening a file |
| FILE_XFER,5 | Error encountered while file was open (i.e., during record access) |
| WARNING,6 | Warning error condition |
| FILE_CLOS,7 | Error related to closing a file |
| FORMAT,8 | Parse error caused by incorrect format |
| INVALID,9 | Invalid DAP field value |
| MSG_SYNC,10 | DAP message received out-of-order |
| > | |
| F RFA,W,3 | Record file address field (1-8) : B |
| F RECNUM2,L | Record number field (1-8) : B |
| F STV,L | Secondary status field (1-8) : B |
| F STX,0 | Descriptor pointing to the |
| | Secondary status text field (1-255) : A |
| F .L,10 | Spare |
| E | |

```

:++
: Define symbols related to the Key Definition Attributes message (TYPE=10).
:--

```

```

$STRUCT DAP,KEYDEF      : DAP key definition Attributes message
F ,L,16                  : Position to message operand section
                          : of DAP control block
F KEYMENU,L              : Key definition menu field (EX-6) : BM
V <M                     : Menu of fields to follow:
  FLG                    : FLG
  DFL                    : DFL
  IFL                    : IFL
  NSG                    : NSG, POS, SIZ
  REF                    : REF
  KNM                    : KNM
  NUL                    : NUL
  IAN                    : IAN
  LAN                    : LAN
  DAN                    : DAN
  DTP                    : DTP
  RVB                    : RVB
  TMP1$,1                : Reserved for HAL
  DVB                    : DVB
  DBS                    : DBS
  IBS                    : IBS
  LVL                    : LVL
  TKS                    : TKS
  MRL                    : MRL
  TMP2$,13               : Reserved
>
K KEYMENU_I,<-            : Define KEYMENU options that are invalid:
  <DAP$M_TMP1$>!-        : Reserved
  <DAP$M_TMP2$>!-        : Reserved
  0>
K KEYMENU_U,<-            : Define KEYMENU options unsupported by VAX:
  0>
F DFL,W                  : Data bucket fill quantity field (2) : B
F IFL,W                  : Index bucket fill quantity field (2) : B
F FLG,B                  : Key options field (EX-3) : BM
V <M                     : Meaning:
  DUP                    : Duplicate key values allowed
  CHG                    : Key field may change on update (alt key)
  NUL CHR                : Null key character defined (alt key)
  TMPT$,5                : Reserved
>
K FLG_I,<-               : Define key options (FLG) that are invalid:
  <DAP$M_TMP1$>!-        : Reserved
  0>
K FLG_U,<-               : Define key options (FLG) unsupported by VAX:
  0>
F NSG,B                  : Number of key segments field (1) : B
F POS,TMP,W              : Temporary work space for POS field processing
S SIZ,TMP,0,B            : Temporary work space for SIZ field processing
F POS,0,8                : Key segment position field (2) : B
S POS0,0,W              : Segment 0
:

```

```

S POS1.2.W
S POS2.4.W
S POS3.6.W
S POS4.8.W
S POS5.10.W
S POS6.12.W
S POS7.14.W

```

```

F SIZ.8.B
S SIZ0.0.B
S SIZ1.1.B
S SIZ2.2.B
S SIZ3.3.B
S SIZ4.4.B
S SIZ5.5.B
S SIZ6.6.B
S SIZ7.7.B

```

```

F KNM.Q

```

```

F REF.B

```

```

F NUL.B

```

```

F IAN.B

```

```

F LAN.B

```

```

F DAN.B

```

```

F DTP.B

```

```

K <

```

```

STG.0

```

```

IN2.1

```

```

BN2.2

```

```

IN4.3

```

```

BN4.4

```

```

PAC.5

```

```

IN8.6

```

```

BN8.7

```

```

>

```

```

K DTP_D,DAPSK_STG

```

```

F MRL.W

```

```

F RVB.L

```

```

F DVB.L

```

```

F DBS.B

```

```

F IBS.B

```

```

F LVL.B

```

```

F TKS.B

```

```

E

```

```

Segment 1

```

```

Segment 2

```

```

Segment 3

```

```

Segment 4

```

```

Segment 5

```

```

Segment 6

```

```

Segment 7

```

```

Key segment size field (1) : B

```

```

Segment 0

```

```

Segment 1

```

```

Segment 2

```

```

Segment 3

```

```

Segment 4

```

```

Segment 5

```

```

Segment 6

```

```

Segment 7

```

```

Descriptor pointing to the

```

```

Key name field (I-40) : A

```

```

Key of reference field (1) : B

```

```

Null key character field (1) : B

```

```

Index area number field (1) : B

```

```

Lowest level index area number field (1) : B

```

```

Data area number field (1) : B

```

```

Key data type field (1) : B

```

```

Data type:

```

```

String

```

```

Signed 2-byte integer

```

```

Unsigned 2-byte integer (binary)

```

```

Signed 4-byte integer

```

```

Unsigned 4-byte integer (binary)

```

```

Packed decimal (0-31 digits plus sign)

```

```

Signed 8-byte integer

```

```

Unsigned 8-byte integer (binary)

```

```

Define default DTP value

```

```

Minimum record length to contain key field (2) : B

```

```

Root bucket start VBN field (I-8) : B

```

```

First data bucket start VBN field (I-8) : B

```

```

Data bucket fill size field (1) : B

```

```

Index bucket fill size field (1) : B

```

```

Level of root buckets field (1) : B

```

```

Total key size field (1) : B

```

```

:++
: Define symbols related to the Allocation Attributes message (TYPE=11).
:--

```

```

$STRUCT DAP,ALLDEF      : DAP Allocation Attributes message
F .L,16                  : Position to message operand section
                          : of DAP control block
F ALLMENU,W              : Allocation menu field (EX-6) : BM
V <M                     : Menu of fields to follow:
  VOL                    : VOL
  ALN                    : ALN
  AOP                    : AOP
  LOC                    : LOC
  TMP1$,1                : Reserved for RFI
  ALQ2                   : ALQ2
  AID                    : AID
  BKZ                    : BKZ
  DEQ2                   : DEQ2
  TMP2$,7                : Reserved
  >
K ALLMENU_I,<-           : Define ALLMENU options that are invalid:
  <DAPSM_TMP1$>!-       : Reserved
  <DAPSM_TMP2$>!-       : Reserved
  0>
K ALLMENU_U,<-           : Define ALLMENU options unsupported by VAX:
  0>
F VOL,W                  : Relative volume number field (2) : B
F ALN,B                  : Alignment options field (EX-4) : BM
K <                      : Alignment types:
  ANY,0                  : Any allocation placement is ok
  CYL,1                  : Align on cylinder boundary
  LBN,2                  : Align on specified logical block
  VBN,3                  : Allocate near specified virtual block
  RFI,4                  : Allocate near specified related file
  >
F AOP,B                  : Allocation options field (EX-4) : BM
V <M                     : Options:
  HRD                    : Return error if requested allocation fails
  CTG2                   : Contiguous space allocation
  CBT2                   : Contiguous-best-try space allocation
  ONC                    : Allocate space on cylinder boundary
  TMP1$,4                : Reserved
  >
K AOP_I,<-               : Define AOP options that are invalid:
  <DAPSM_TMP1$>!-       : Reserved
  0>
K AOP_U,<-               : Define AOP options unsupported by VAX:
  0>
F .B,2                   : Padding
F LOC,L                  : Starting location field (I-8) : B
F ALQ2,L                 : Allocation quantity field (I-5) : B
F AID,B                  : Area identification field (1) : B
F BKZ,B                  : Bucket size field (1) : B
F DEQ2,W                 : Default extension quantity field (2) : B
F .L,11                  : Spare

```

DAPDEF.MDL;1

16-SEP-1984 16:38:17.80 ¹⁶ Page 28

E

:

```

:++
: Define symbols related to the Summary Attributes message (TYPE=12).
:--

```

```

$STRUCT DAP.SUMDEF      : DAP summary Attributes message
F .L,16                  : Position to message operand section
                          : of DAP control block
F SUMENU,W               : Summary menu field (EX-6) : BM
V <M                     : Menu of fields to follow:
  NOK                     : NOK
  NOA                     : NOA
  NOR                     : NOR
  PVN                     : PVN
  TMP1$,12               : Reserved
>
K SUMENU_I,<-             : Define SUMENU options that are invalid:
  <DAP$M_TMP1$>!-       : Reserved
  0>
K SUMENU_U,<-             : Define SUMENU options unsupported by VAX:
  0>                     : Note: allow NOR
F PVN,W                  : Prologue version number field (1) : B
F NOK,B                  : Number of keys field (1) : B
F NOA,B                  : Number of allocation areas field (1) : B
F NOR,B                  : Number of record descriptors field (1) : B
F .B                     : Padding
F .L,14                  : Spare
E

```

```

++
: Define symbols related to the Date and Time Attributes message (TYPE=13).
--

```

```

$STRUCT DAP,TIMDEF      : DAP date and time Attributes message
F .L,16                  : Position to message operand section
                          : of DAP control block
F TIMENU,W               : Date and time menu field (EX-6) : BM
V <M                     : Menu of fields to follow:
  CDT                    : CDT
  RDT                    : RDT
  EDT                    : EDT
  RVN                    : RVN
  BDT                    : BDT
  PDT                    : PDT
  ADT                    : ADT
  TMP1$,9                : Reserved
>
K TIMENU_I,<-             : Define TIMENU options that are invalid:
  <DAPSM_TMP1$>!-       : Reserved
0>
K TIMENU_U,<-             : Define TIMENU options unsupported by VAX:
  0>                     : Note: allow PDT and ADT
F RVN,W                  : Revision number field (2) : B
F .L                      : Padding
F CDT,Q                  : Creation date and time field (18) : A
                          : (stored in DAP control block as a
                          : 64-bit time value per VMS convention)
F RDT,Q                  : Revision date and time field (18) : A
                          : (stored in DAP control block as a
                          : 64-bit time value per VMS convention)
F EDT,Q                  : Expiration date and time field (18) : A
                          : (stored in DAP control block as a
                          : 64-bit time value per VMS convention)
F BDT,Q                  : Backup date and time field (18) : A
                          : (stored in DAP control block as a
                          : 64-bit time value per VMS convention)
F PDT,Q                  : Physical creation date and time field (18) : A
                          : (stored in DAP control block as a
                          : 64-bit time value per VMS convention)
F ADT,Q                  : Accessed date and time field (18) : A
                          : (stored in DAP control block as a
                          : 64-bit time value per VMS convention)
F .L,2                   : Spare
E

```

```

**
: Define symbols related to the Protection Attributes message (TYPE=14).
--

```

```

$STRUCT DAP,PRODEF      : DAP protection Attributes message
F ,L,16                  : Position to message operand section
                          : of DAP control block
F PROMENU,W              : Protection menu field (EX-6) : BM
  V <M                   : Menu of fields to follow:
    OWNER                : OWNER
    PROSYS                : PROSYS
    PROOWN                : PROOWN
    PROGRP                : PROGRP
    PROWLD                : PROWLD
    TMP1$,11             : Reserved
  >
  K PROMENU_I,<-          : Define PROMENU options that are invalid:
    <DAPSM_TMP1$>!-      : Reserved
    0>
  K PROMENU_U,<-          : Define PROMENU options unsupported by VAX:
    0>
F ,W,3                   : Padding
F OWNER,Q                : Descriptor pointing to the
                          : File owner field (I-40) : A
F PROSYS,W               : System protection field (EX-3) : BM
  V <M                   : Meaning:
    RED_ACC               : Deny read access
    WRT_ACC               : Deny write access
    EXE_ACC               : Deny execute access
    DLT_ACC               : Deny delete access
    APP_ACC               : Deny append access
    DIR_ACC               : Deny directory access
    UPD_ACC               : Deny update access
    CHG_ACC               : Deny change protection access
    EXT_ACC               : Deny extend access
    TMPT$,7              : Reserved
  >
  K PROTECT_I,<-          : Define protection options that are invalid:
    <DAPSM_TMP1$>!-      : Reserved
    0>
    This mask applies to PROSYS, PROOWN, PROGRP,
    and PROWLD fields
  K PROTECT_U,<-          : Define protection options unsupported by VAX:
    0>
    This mask applies to PROSYS, PROOWN, PROGRP,
    and PROWLD fields
    Note: allow APP_ACC, DIR_ACC, UPD_ACC,
           CHG_ACC, and EXT_ACC
F PROOWN,W               : Owner protection field (EX-3) : BM
F PROGRP,W               : Group protection field (EX-3) : BM
F PROWLD,W               : World protection field (EX-3) : BM
F ,L,10                  : Spare
E

```

```

:++
: Define symbols related to the Name Attributes message (TYPE=15).
:--

```

```

$STRUCT DAP,NAMDEF      : DAP name Attributes message
F ,L,16                  : Position to message operand section
                          : of DAP control block
F NAMETYPE,B            : Name type field (EX-3) : BM
  V <M                   : Type:
    FILSPEC              : Primary file specification
    FILNAME              : File name
    DIRNAME              : Directory name
    VOLNAME              : Volume or structure name
    DFISPEC              : Default file specification
    TMP1$,1              : Reserved for RELSPEC
    TMP2$,2              : Reserved
  >
F ,B,3                   : Padding
  K NAMETYP I,<-          : Define NAMETYPE options that are invalid:
    <DAPSM_TMP1$>!--      : Reserved
    <DAPSM_TMP2$>!--      : Reserved
    0>
  K NAMETYP U,<-          : Define NAMETYPE options unsupported by VAX:
    <DAPSM_DFISPEC>!--    : DFISPEC
    0>
F NAMESPEC,0            : Descriptor pointing to the
                          : Name field (I-255) : A
F ,L,13                  : Spare
E

```

++
: Define symbols related to DAP message CRC checksum computation.
: The CRC polynomial function (order 16) used is:

--
: $X^{16} + X^{15} + X^{13} + X^7 + X^4 + X^2 + X + 1$
:

: \$STRUCT DAP,CRCDEF : DAP message CRC checksum symbol definitions

: K CRC_INIT,<^X0000FFFF> : Initial CRC value
: K CRC_POLY,<^X0000E905> : CRC polynomial representation used as
: : input to LIB\$CRC_TALBE to generate
: : the CRC polynomial table below:

: K CRC_TBL0,<^X00000000> : Table entry 0
: K CRC_TBL1,<^X000053E3> : Table entry 1
: K CRC_TBL2,<^X0000A7C6> : Table entry 2
: K CRC_TBL3,<^X0000F425> : Table entry 3
: K CRC_TBL4,<^X00009D87> : Table entry 4
: K CRC_TBL5,<^X0000CE64> : Table entry 5
: K CRC_TBL6,<^X00003A41> : Table entry 6
: K CRC_TBL7,<^X000069A2> : Table entry 7
: K CRC_TBL8,<^X0000E905> : Table entry 8
: K CRC_TBL9,<^X0000BAE6> : Table entry 9
: K CRC_TBLA,<^X00004EC3> : Table entry 10
: K CRC_TBLB,<^X00001D20> : Table entry 11
: K CRC_TBLC,<^X00007482> : Table entry 12
: K CRC_TBLD,<^X00002761> : Table entry 13
: K CRC_TBLE,<^X0000D344> : Table entry 14
: K CRC_TBLF,<^X000080A7> : Table entry 15
: E

```

:++
:SDAPFIDDEF defines DAP field identification code symbols.
:These are used to identify a field in a DAP Status message.
:--

```

```

$STRUCT DAP,FIDDEF      : DAP field ID codes

K <,$                   : Miscellaneous field codes:
  UNKNOWN,0              :   Unknown field
  TYPE,8                  :   DAP message type field
>
K <,$                   : Message header field codes:
  FLAGS,8                 :   DAP message flags field
  STREAMID,9              :   Data stream identification field
  LENGTH,10               :   Length field
  LEN256,11               :   Length extension field
  BITCNT,12               :   Bit count field
                           :   (13) reserved
  SYSPEC,14               :   System specific field
                           :   whose subfields use the same code:
  SSP_MENU,14             :   System specific menu field
  SSP_CAP,14              :   System specific capabilities field
  SSP_FLG,14              :   System specific flags field
>
K <,$                   : Configuration message field codes:
  BUFSIZ,16               :   Buffer size field
  OSTYPE,17               :   Operating system type field
  FILESYS,18              :   File system type field
  VERNUM,19               :   DAP version number field
  ECONUM,20               :   ECO version number field
  USRNUM,21               :   User protocol version number field
  DECVER,22               :   DEC software version number field
  USRVER,23               :   User software version number field
  SYSCAP,24               :   System capabilities field
>
K <,$                   : Attributes message field codes:
  ATTRMENU,16             :   Attributes menu field
  DATATYPE,17             :   Data type field
  ORG,18                  :   File organization field
  RFM,19                  :   Record format field
  RAT,20                  :   Record attributes field
  BLS,21                  :   Block size field
  MRS,22                  :   Maximum record size field
  ALQ1,23                 :   Allocation quantity field
  BKS,24                  :   Bucket size field
  FSZ,25                  :   Fixed control area size field
  MRN,26                  :   Maximum record number field
  RUNSYS,27               :   Run-time system field
  DEQ1,28                 :   Default extension quantity field
  FOP1,29                 :   File options field
  BSZ,30                  :   Byte size field
  DEV,31                  :   Device characteristics field
                           :   (32) reserved for SDC field
  LRL,33                  :   Longest record length field
  HBK,34                  :   Highest virtual block number field
  EBK,35                  :   End-of-file block number field

```

| | | |
|-------------|---|--|
| FFB,36 | : | First free byte in EOF block field |
| SBN,37 | : | Starting logical block number field |
| > | : | |
| K <,\$ | : | Access message field codes: |
| ACCFUNC,16 | : | Access function field |
| ACCOPT,17 | : | Access options field |
| FILESPEC,18 | : | File specification field |
| FAC,19 | : | File access field |
| SHR,20 | : | File sharing field |
| DISPLAY1,21 | : | Display attributes field |
| PASSWORD,22 | : | Password field |
| > | : | |
| K <,\$ | : | Control message field codes: |
| CTLFUNC,16 | : | Control function field |
| CTLMENU,17 | : | Control menu field |
| RAC,18 | : | Record access field |
| KEY,19 | : | Key field |
| KRF,20 | : | Key of reference field |
| ROP,21 | : | Record options field |
| | : | (22) reserved for HSH field |
| DISPLAY2,23 | : | Display attributes field |
| BLKCNT,24 | : | Block count field |
| > | : | |
| K <,\$ | : | Continue Transfer message field codes: |
| CONFUNC,16 | : | Continue transfer function field |
| > | : | |
| | : | Acknowledge message field codes: |
| | : | none |
| K <,\$ | : | Access Complete message field codes: |
| CMPPFUNC,16 | : | Access complete function field |
| POP2,17 | : | File options field |
| CHECK,18 | : | CRC Checksum field |
| > | : | |
| K <,\$ | : | Data message field codes: |
| RECRUM1,16 | : | Record number field |
| FILEDATA,17 | : | File data field |
| > | : | |
| K <,\$ | : | Status message field codes: |
| SISCODE,16 | : | Status code field used for both: |
| | : | MACCODE,16 |
| | : | MICCODE,17 |
| RFA,18 | : | Record file address field |
| RECRUM2,19 | : | Record number field |
| STV,20 | : | Secondary status value field |
| STX,21 | : | Secondary status text field |
| > | : | |
| K <,\$ | : | Key definition attributes message field codes: |
| KEYMENU,16 | : | Key definition menu field |
| FLG,17 | : | Key options field |
| DFL,18 | : | Data bucket fill quantity field |
| IFL,19 | : | Index bucket fill quantity field |
| NSG,20 | : | Number of key segments field |
| POS,21 | : | Key segment position field |
| POS_TMP,21 | : | (alias for POS) |
| SIZ,22 | : | Key segment size field |
| SIZ_TMP,22 | : | (alias for SIZ) |

| | | |
|-------------|---|---|
| REF,23 | : | Key of reference field |
| KNM,24 | : | Key name field |
| NUL,25 | : | Null key character field |
| IAN,26 | : | Index area number field |
| LAN,27 | : | Lowest level index area number field |
| DAN,28 | : | Data area number field |
| DTP,29 | : | Key data type field |
| RVB,30 | : | Root bucket start VBN field |
| | : | (31) reserved for HAL field |
| DVB,32 | : | First data bucket start VBN field |
| DBS,33 | : | Data bucket fill size field |
| IBS,34 | : | Index bucket fill size field |
| LVL,35 | : | Level of root buckets field |
| TKS,36 | : | Total key size field |
| MRL,37 | : | Minimum record length to contain key field |
| > | : | |
| K <,\$ | : | Allocation attributes message field codes: |
| ALLMENU,16 | : | Allocation menu field |
| VOL,17 | : | Relative volume number field |
| ALN,18 | : | Alignment options field |
| AOP,19 | : | Allocation options field |
| LOC,20 | : | Starting location field |
| | : | (21) reserved for RFI field |
| ALQ2,22 | : | Allocation quantity field |
| AID,23 | : | Area identification field |
| BKZ,24 | : | Bucket size field |
| DEQ2,25 | : | Default extension quantity field |
| > | : | |
| K <,\$ | : | Summary attributes message field codes: |
| SUMENU,16 | : | Summary menu field |
| NOK,17 | : | Number of keys field |
| NOA,18 | : | Number of allocation areas field |
| NOR,19 | : | Number of record descriptors field |
| PVN,20 | : | Prologue version number field |
| > | : | |
| K <,\$ | : | Date and time attributes message field codes: |
| TIMENU,16 | : | Date and time menu field |
| CDT,17 | : | Creation date and time field |
| RDT,18 | : | Revision date and time field |
| EDT,19 | : | Expiration date and time field |
| RVN,20 | : | Revision number field |
| BDT,21 | : | Backup date and time field |
| PDT,22 | : | Physical creation date and time field |
| ADT,23 | : | Accessed date and time field |
| > | : | |
| K <,\$ | : | Protection attributes message field codes: |
| PROMENU,16 | : | Protection menu field |
| OWNER,17 | : | File owner field |
| PROSYS,18 | : | System protection field |
| PROOWN,19 | : | Owner protection field |
| PROGRP,20 | : | Group protection field |
| PROWLD,21 | : | World protection field |
| > | : | |
| K <,\$ | : | Name message field codes: |
| NAMETYPE,16 | : | Name type field |
| NAMESPEC,17 | : | Name field |

E >

:

: End of module

FA

BU
LA

LA

+

0173 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SCHFCB
LIS

SND5MB
LIS

SHFDIR
LIS

SNDER
LIS

TRUNC
LIS

FAL

FAL
MAP

SELVOL
LIS

DAPDEF
MOL

SMALOC
LIS

SNOBAD
LIS

SWTTL
LIS

WITURN
LIS

0174 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

